

INSIDE DOS

Tips & techniques for MS-DOS & PC-DOS Versions 5 and 6

VERSION
6.0

An easy way to make CHOICES with a batch file

One obvious shortcoming of DOS batch files in versions prior to 6 is that they don't provide a direct way to solicit replies from the user—they aren't interactive. In past issues of *Inside DOS* ("Soliciting Input in a Batch File," September 1990; "A Better Way to Solicit Input in a Batch File," December 1990; and "Soliciting a Reply," March 1992), we showed you how to create DEBUG scripts to work around this shortcoming. (For a review of one of these scripts, read "The Most Efficient Way to Solicit User Input in DOS 5 Batch Files," on page 3.) The problem with DOS DEBUG scripts is that they're often difficult to interpret and, therefore, difficult to troubleshoot.

DOS 6 does away with the need to use DEBUG scripts for interactive batch files by introducing the new CHOICE command. CHOICE, while not a highly publicized addition to DOS, might be one that deserves the most esteem. This command lets a batch file display a customized prompt, specify a set of appropriate responses, and select a default response if the user doesn't select one within a given time period—without using files or commands external to the batch file. In this article, we'll show you how easy CHOICE makes soliciting user input in a DOS 6 batch file.

The CHOICE syntax

Anytime you want the user to respond to a batch file prompt by making a selection, you simply add the CHOICE command in the form

```
choice text /C:keys
```

where *text* is the message explaining what input the batch file wants and *keys* are the allowable keys the user can press to respond. If you don't specify *text*, CHOICE simply displays the valid keys the user can press. When CHOICE displays the user's choices, it places them between brackets, separates them with commas, and follows the closing bracket with a question mark. If you omit the /C switch, CHOICE will allow Y and N as valid choices.

For instance, suppose you're creating a menu system and you want the user to press a letter to run a

program. Your CHOICE command could look something like

```
choice Which program do you want to run /C:wesc
```

When you run the batch file, this CHOICE command will display

```
Which program do you want to run [W,E,S,C]?
```

If you want to present a multiline prompt, you can use the CHOICE command to display the last line of the prompt and precede it with ECHO commands that display the previous lines, as in

```
echo W - Microsoft Windows
echo E - MS-DOS Editor
echo S - MS-DOS Shell
echo C - command prompt
echo.
choice Which program do you want to run /C:wesc
```

Setting a response time limit

When you want to limit the length of time in which the user can respond to a prompt, you can use the /T switch

IN THIS ISSUE

- An easy way to make CHOICES with a batch file 1
- The most efficient way to solicit user input in DOS 5 batch files 3
- DOS 6's UNDELETE offers three levels of protection from file deletion 4
- Undeleting files in DOS 5 6
- Other switches you can use with UNDELETE 8
- Check your PATH before running MemMaker 9
- Van Wolverton: DOS 6 won't lose your data 10
- Nothing—not even DoubleSpace—is forever 10
- Tell us about yourself 12

with CHOICE. The /T switch causes CHOICE to default to a specified key if the user doesn't respond within the specified time period. With the /T switch, the syntax of CHOICE is

```
choice text /C:keys /T:defkey,secs
```

where *defkey* is the default choice from among the valid keys specified by *keys* and *secs* is the number of seconds—from 0 to 99—CHOICE will wait for a response from the user. In the previous example, if you want CHOICE to default to the command prompt if the user doesn't select a program within 30 seconds, you'd use the following CHOICE command:

```
choice Which program do you want to run /C:wesc /T:c,30
```

Processing the response to a CHOICE prompt

When DOS processes a CHOICE command, it assigns the first valid key a value of 1, the second valid key a value of 2, and so forth. When the user presses a key, DOS stores the value of the pressed key in the system variable ERRORLEVEL. If the user presses an invalid key, DOS tells the computer to sound a warning beep. If the user presses [Ctrl][Break] or [Ctrl]C, DOS stores the value 0 in ERRORLEVEL.

To make the batch file carry out a series of instructions based on the user's choice, you follow the CHOICE

command with a series of IF commands that checks the value of ERRORLEVEL and branches to the appropriate section of the batch file. The syntax of the IF command that checks the value of ERRORLEVEL and branches to a section of the batch file is

```
if ERRORLEVEL v goto LABEL
```

where *v* is the value you're comparing ERRORLEVEL to and *LABEL* is the label that marks the section you want to branch to if ERRORLEVEL matches *v*. For instance, to process the CHOICE command in the previous example, you might follow it with

```
if ERRORLEVEL 4 goto :END
if ERRORLEVEL 3 goto :SHELL
if ERRORLEVEL 2 goto :EDITOR
if ERRORLEVEL 1 goto :WINDOWS
```

Note that you must check the value of ERRORLEVEL from highest to lowest because DOS interprets this IF command as *if ERRORLEVEL is greater than or equal to v, branch to LABEL*. Therefore, by checking for the highest possible value of ERRORLEVEL first, DOS bypasses the other IF statements by branching past them as soon as it finds a match.

A CHOICE example

To see firsthand how the CHOICE command works in a batch file, create the PROGMENU.BAT file shown in Fig-

INSIDE DOS

Inside DOS (ISSN 1049-5320) is published monthly by The Cobb Group.

Prices	Domestic:	\$49/yr. (\$6.00 each)
	Outside US:	\$69/yr. (\$8.50 each)

Phone	Toll free:	(800) 223-8720
	Local:	(502) 491-1900
	Customer Relations Fax:	(502) 491-8050
	Editorial Department Fax:	(502) 491-4200

Address You may address tips, special requests, and other correspondence to

The Editor, Inside DOS
9420 Bunsen Parkway, Suite 300
Louisville, KY 40220

For subscriptions, fulfillment questions, and requests for bulk orders, address your letters to

Customer Relations
9420 Bunsen Parkway, Suite 300
Louisville, KY 40220

Postmaster Second class postage is paid in Louisville, KY. Send address changes to

Inside DOS
P.O. Box 35160
Louisville, KY 40232

Back Issues

To order back issues, call Customer Relations at (800) 223-8720. Back issues cost \$6.00 each, \$8.50 outside the US. You can pay with MasterCard, VISA, Discover, or American Express, or we can bill you. Please identify the issue you want by the month and year it was published. Customer Relations can also provide you with an issue-by-issue listing of all the articles that have appeared in Inside DOS.

Copyright

Copyright © 1993, The Cobb Group. All rights are reserved. Inside DOS is an independently produced publication of The Cobb Group. The Cobb Group reserves the right, with respect to submissions, to revise, republish, and authorize its readers to use the tips submitted for both personal and commercial use. The Cobb Group, its logo, and the Satisfaction Guaranteed statement and seal are registered trademarks of The Cobb Group. Inside DOS is a trademark of The Cobb Group. Microsoft and MS-DOS are registered trademarks and Microsoft Windows is a trademark of Microsoft Corporation. Apple and Macintosh are registered trademarks of Apple Computers, Inc. Artisoft and LANtastic are registered trademarks of Artisoft, Inc. CompuServe Mail is a service mark of CompuServe Inc. LaserMaster is a registered trademark of LaserMaster Corporation. MCI Mail is a registered trademark of MCI Communications Corporation.

Staff

Editor-in-Chief:	Charity Edelen
Contributing Editors:	Van Wolverton
	David Reid
	Elizabeth Welch
Editing:	Timothy E. Hampton
	Julie Jefferson
	Karl Feige
Production Artist:	Suzanne Thornberry
Design:	Tara Dickerson
Managing Editor:	Brent Shean
Publications Manager:	Linda Baughman
Circulation Manager:	Jeff Yocom
Publications Director:	Mark Crane
Editorial Director:	Jon Pyles
Publishers:	

Advisory Board

Earl Berry Jr.
Tina Covington
Marvin D. Livingood

ure A on page 4. You can create this batch file in the DOS Editor and store it in the C:\BATCH directory by typing

```
C:\>edit c:\batch\progmnu.bat
```

to open the Editor and then typing the commands just as they appear in Figure A. (Since EDIT is an external command stored in the C:\DOS directory, make sure that C:\DOS is in your path or switch to the C:\DOS directory before issuing the EDIT command.) After typing the batch file, press [Alt]F,X to close the Editor. Before closing, the Editor will prompt you to save the file. When it does, press Y to return to the DOS prompt.

Assuming the C:\BATCH directory is in your path, type

```
C:\>progmnu
```

to run the PROGMENU.BAT file. The first line

```
@echo off
```

tells DOS not to display the batch file commands as it executes them. The next line

```
cls
```

The most efficient way to solicit user input in DOS 5 batch files

VERSION
5.0

Of the three DEBUG programs we've published that solicit user input (KEYPRESS.COM, Version 1, in September 1990; KEYPRESS.COM, Version 2, in December 1990; and REPLY.COM in March 1992), REPLY.COM is probably the most efficient program for the job. The REPLY.COM program comes from Van Wolverton's book *Supercharging MS-DOS*, published by Microsoft Press. For our readers who are waiting to upgrade to DOS 6 but would still like to create interactive batch files, we'll review the REPLY.COM program.

REPLY.COM accepts any key or valid key combination (such as [Ctrl]J or [Alt][F10]) and sets the system variable ERRORLEVEL to the key code of the pressed key(s). To create REPLY.COM, you first create a script file containing commands for the DOS DEBUG program. Then, you run DEBUG and redirect its input to the script file. When DEBUG carries out the commands in the script file, it creates REPLY.COM and then returns to DOS.

Creating the script file

A DEBUG script file must contain only ASCII characters. You can create the script file, REPLY.SCR, with the DOS Editor or with a word processor that allows you to store a text file without special formatting codes. However, the quickest way to create REPLY.SCR is to copy lines from the console to the file. Once you've entered a line, you can't go back and change it, so make sure each line is correct before you press [Enter].

To create REPLY.SCR in this fashion, type the following lines exactly as shown:

```
C:\>copy con reply.scr
e100 b4 08 cd 21 3c 00 75 02 cd 21 b4 4c cd 21
rcx
e
w
q
```

Now, press [F6] and then [Enter]. DOS will respond with *1 file(s) copied*.

Creating REPLY.COM

Creating REPLY.COM from the script file takes even less time. A single command starts DEBUG, names the file that you're creating (REPLY.COM), and tells DOS to redirect its input from the keyboard to the file named REPLY.SCR:

```
C:\>debug reply.com < reply.scr
```

When you type this command and press [Enter], DEBUG creates REPLY.COM and then returns to the DOS prompt.

Testing REPLY.COM

To make sure REPLY.COM is working correctly and doing something useful, you must use it in a batch file. Type the following to create a simple test file named TEST.BAT:

```
C:\>copy con test.bat
@echo off
echo Press Ctrl-K
:Wait
reply
if errorlevel 11 if not errorlevel 12 goto :END
goto WAIT
:END
```

After you've typed the file, press [F6] and then [Enter].

This batch file displays the message *Press Ctrl-K* and waits for you to press a key. If you press [Ctrl]K, it returns to DOS; if you press any other key, the batch file continues to wait. The key code for [Ctrl]K is 11, so the IF ERRORLEVEL command makes sure that the ERRORLEVEL returned by REPLY.COM is 11 or more but not 12 or more; only 11 satisfies the two conditions.

clears the screen. The ECHO and CHOICE commands display on the screen

SELECT-A-PROGRAM MENU

W - Microsoft Windows
E - MS-DOS Editor
S - MS-DOS Shell
C - command prompt

Which program do you want to run [W,E,S,C]?

Figure A

```
@echo off
cls
echo SELECT-A-PROGRAM MENU
echo.
echo W - Microsoft Windows
echo E - MS-DOS Editor
echo S - MS-DOS Shell
echo C - command prompt
echo.
choice Which program do you want to run /C:wesc
if ERRORLEVEL 4 goto :END
if ERRORLEVEL 3 goto :SHELL
if ERRORLEVEL 2 goto :EDITOR
if ERRORLEVEL 1 goto :WINDOWS
:SHELL
dosshell
goto :END
:EDITOR
edit
goto :END
:WINDOWS
win
:END
cls
```

The PROGMENU.BAT file lets you start a program by selecting it from a menu.

If you press C to select the command prompt, DOS stores the value 4 in ERRORLEVEL. This value meets the condition for the first IF statement, which branches to the :END section of the batch file. This section simply clears the screen again and returns to the DOS prompt, ending the batch file.

If you press S to select the MS-DOS Shell, DOS stores the value 3 in ERRORLEVEL. This value meets the condition for the second IF statement, which then branches to the :SHELL section of the batch file. This section issues the DOSHELL command to open the DOS Shell. Once you exit the Shell, the batch file resumes and executes the instructions in the :END section.

If you press E to select the MS-DOS Editor, DOS stores the value 2 in ERRORLEVEL. This value meets the condition for the third IF statement, which branches to the :EDITOR section of the batch file. The :EDITOR section issues the EDIT command to open the DOS Editor. Once you exit the Editor, the batch file resumes and branches to the :END section.

If you press W to select Microsoft Windows, DOS stores the value 1 in ERRORLEVEL. This value meets the condition for the last IF statement, which branches to the :WINDOWS section of the batch file. This section issues the WIN command to start Windows. Once you exit Windows, the batch file resumes and executes the instructions in the :END section.

Conclusion

Creating a DEBUG script to make your batch files interactive might seem like more trouble than it's worth—and in DOS 6, it is. The latest version of DOS gives you the CHOICE command to accomplish what, in past versions, only a DEBUG script could—solicit user input. ■

SAFETY FEATURES

VERSION
6.0

DOS 6's UNDELETE offers three levels of protection from file deletion

We all know how easy it is to delete a file. You simply type *del* followed by the file specification and—presto!—the file disappears. In fact, deleting a file is dangerously easy. Since DOS doesn't ask you to confirm deletions unless you're deleting all the files in a directory, a mere typing error can mean the loss of files you had no intention of deleting. Fortunately, if you catch your error immediately, you can use DOS' UNDELETE utility to recover these accidentally deleted files.

In the January 1992 *Inside DOS* ("A Second Chance: How DOS 5 'Undeletes' Files"), we introduced the UNDELETE utility. (For a recap, read "Undeleting Files in DOS 5," on page 6.) In this article, we'll explain how to use DOS 6's version of UNDELETE. We'll start by reviewing how DOS stores files and how it deletes them.

How DOS stores files

To store a file, DOS allocates disk space in clusters. DOS might allocate several clusters for the file based on the

file's size. The clusters may occur consecutively or they might spread out across the disk, depending on the available disk space. DOS keeps track of the sequence of clusters that store a file by using a File Allocation Table (FAT). If a cluster is unoccupied, the FAT records a 0 status for the cluster. If the cluster contains file information, the FAT records the number of the next cluster, if one follows in the sequence. Otherwise, the FAT records an end-of-file status for the cluster.

A directory—what usually comes to mind when we think of where a file is stored—is actually just a list containing each file's name, size, date and time stamp, and starting cluster number. When you access a file, DOS searches the specified directory for a matching filename. When DOS finds a matching entry, it uses the file's starting cluster number to locate the file on disk by following the cluster sequence recorded in the FAT.

How DOS deletes files

When you issue the DEL command to delete a file from a disk, DOS doesn't actually erase the file's contents from the disk. Instead, DOS simply changes to 0 the status of the clusters the file currently occupies to indicate that the clusters are now available for another file. At this point, the file is still recoverable. However, if you delete a file and then create another, DOS might store the new file's contents in one or more of the deleted file's clusters. If it does, you won't be able to recover the deleted file.

Undeleting files in DOS 6

DOS provides the UNDELETE utility to restore any files you just deleted with the DEL command. By default, UNDELETE lets you recover any deleted file as long as DOS hasn't stored another file in the deleted file's clusters. However, you can configure UNDELETE to provide higher degrees of file-deletion protection. Let's first look at UNDELETE's standard mode.

Standard delete protection

To restore a file by using UNDELETE's standard mode, you simply type

```
C:\>undelete drive:\path\filename
```

If you don't specify a drive or path, UNDELETE assumes you deleted the file from the current directory. You can use wildcards (* or ?) in the filename to restore several files, or you can omit all the parameters to restore all the deleted files in the current directory that can be restored.

Suppose, for example, you want to delete all the files in your C:\BATCH directory that have a BAK extension. However, instead of typing

```
C:\>del c:\batch\*.bak
```

you typed

```
C:\>del c:\batch\*.bat
```

To recover all the batch files you just deleted, you type

```
C:\>undelete c:\batch\*.bat
```

After displaying some copyright information, the UNDELETE command displays

```
Directory: C:\BATCH
File Specifications: *.BAT
```

```
Delete Sentry control file not found.
```

```
Deletion-tracking file not found.
```

```
MS-DOS directory contains    x deleted files.
Of those,    x files may be recovered.
```

Using the MS-DOS directory method.

```
?ilename BAT  size date time ...attrib Undelete (Y/N)?
```

At this prompt, you press Y, and DOS responds

```
Please type the first character for ?ilename.BAT: _
```

After you type the first character in the filename, DOS responds

```
File successfully undeleted.
```

Then, DOS displays information for the next file and asks if you want to undelete the file. DOS repeats this sequence of prompts for each recoverable file. Once you've responded to the prompts for each file, DOS returns you to the command prompt.

The Delete Tracker protection level

If you want a degree of deletion protection that's a step above the standard mode, try the Delete Tracker protection level. (This level of protection is similar to the level DOS 5's MIRROR program offers.) Delete Tracker records the file cluster numbers a file occupies when you delete it and saves this information in a hidden file called PCTRACKR.DEL. Therefore, even if DOS saves another file in one of the clusters a deleted file occupies, you might still be able to recover part of the file.

To configure UNDELETE to operate in Delete Tracker mode, you type

```
C:\>undelete /t drive
```

where *drive*, a required parameter, is the letter of the drive for which you want Delete Tracker protection. For instance, if you want to enable Delete Tracker protection

for the C: drive, you type

```
C:\>undelete /tc
```

After displaying copyright information, this command responds

```
UNDELETE loaded.
```

```
Delete Protection Method is Delete Tracking.  
Enabled for drives : C
```

If you want Delete Tracker protection enabled all the time, include this command in your AUTOEXEC.BAT file. Once you invoke Delete Tracker protection, typing

```
C:\>undelete drive:\path\filename
```

recovers a deleted file by retrieving its cluster numbers from the PCTRACKR.DEL file and restoring the cluster sequence in the FAT. If you'd invoked Delete Tracker protection before deleting the *.BAT files in our previous example, the response to

```
C:\>undelete c:\batch\*.bat
```

after the preliminary copyright information would have been

```
Directory: C:\BATCH  
File Specifications: *.BAT
```

```
Delete Sentry control file not found.
```

```
Deletion-tracking file contains  x deleted files.  
Of those,      x files have all clusters available,  
               0 files have some clusters available,  
               0 files have no clusters available.
```

```
MS-DOS directory contains  x deleted files.  
Of those,      x files may be recovered.
```

```
Using the Deletion-tracking method.
```

```
filename BAT size date time ...attrib Deleted: date time  
All of the clusters for this file are available. Undelete  
(Y/N)?
```

At this prompt, you press Y, and DOS responds

```
File successfully undeleted.
```

Then, DOS displays information for the next file and asks whether you want to undelete the file. DOS repeats this sequence for each recoverable file. Once you've re-

Undeleting files in DOS 5

Although Version 5's UNDELETE utility isn't as flexible as the version we describe in "DOS 6's UNDELETE Offers Three Levels of Protection from File Deletion," the earlier version is still a valuable utility for recovering recently deleted files. DOS 5's UNDELETE offers two levels of deletion protection, the higher of which involves the MIRROR program.

Standard delete protection

Using default deletion protection, you type

```
C:\>undelete drive:\path\filename
```

and press [Enter] to recover a file you just deleted with the DEL command. If you don't specify a drive or path, UNDELETE assumes you deleted the file from the current directory. You can use wildcards (* or ?) in the filename to restore several files, or you can omit all the parameters to restore all the deleted, but recoverable, files in the current directory.

For instance, suppose you accidentally deleted all the batch files in your C:\BATCH directory. To recover them, you type

```
C:\>undelete c:\batch\*.bat
```

The UNDELETE command responds

```
Directory: C:\BATCH  
File Specifications: *.BAT
```

```
Deletion-tracking file not found.
```

```
MS-DOS directory contains  x deleted files.  
Of those,      x files may be recovered.
```

```
Using the MS-DOS directory.
```

```
?ilename BAT size date time ...attrib Undelete (Y/N)?
```

At this prompt, you press Y, and DOS responds

```
Please type the first character for ?ilename.BAT: _
```

After you type the first character, DOS responds

```
File successfully undeleted.
```

Then, DOS displays information for the next file and asks whether you want to undelete the file. DOS repeats this sequence of prompts for each recoverable file. Once you've responded to the prompts for each file, DOS returns you to the command prompt.

VERSION
5.0

sponded to the prompt for each file, DOS returns you to the command prompt.

Although Delete Tracker offers a higher degree of deletion protection than UNDELETE's standard mode, it requires 13.5 Kb of memory for the memory-resident portion of the UNDELETE utility. Delete Tracker also takes up a minimal amount of disk space to store the PCTRACKR.DEL file.

The Delete Sentry protection level

The highest level of protection UNDELETE offers is Delete Sentry. Delete Sentry creates a hidden directory called SENTRY. When you delete a file with Delete Sentry enabled, DOS moves the file from its current directory to the SENTRY directory without changing the cluster sequence recorded in the FAT. Since the FAT remains unchanged, DOS won't store new files in the same clusters as the deleted files, making full file recovery a snap. If you choose to undelete the file, DOS simply moves it back to its original directory.

To configure UNDELETE in Delete Sentry mode, you type

```
C:\>undelete /sdrive
```

where *drive* is the letter of the drive for which you want Delete Sentry protection. The *drive* parameter in this instance is optional. If you don't specify a drive, you invoke Drive Sentry protection for the current drive. For instance, if C: is the current drive and you type

```
C:\>undelete /s
```

you'll see the preliminary copyright information followed by

```
UNDELETE loaded.
```

```
Delete Protection Method is Delete Sentry.  
Enabled for drives : C
```

If you want Delete Sentry protection enabled all the time, include this command in your AUTOEXEC.BAT file.

Once you invoke Delete Sentry protection, you move a deleted file from the SENTRY directory to its original directory by typing

```
C:\>undelete drive:\path\filename
```

where *drive* and *path* are the original drive and path of the deleted file. For example, if you'd invoked Delete

The Delete Tracker protection level

In addition to the default UNDELETE mode, DOS 5 provides the MIRROR command, which enables file deletion-tracking. When you activate the MIRROR command, DOS records, in a deletion-tracking file, the directory and File Allocation Table (FAT) entries for each file it deletes. If you later need to undelete a file, UNDELETE uses the information in MIRROR's tracking file to recover deleted files.

To enable file deletion-tracking, you simply type

```
C:\>mirror /tdrive
```

For instance, if you want to enable file deletion-tracking for the C: drive, you type

```
C:\>mirror /tc
```

MIRROR responds with

Creates an image of the system area.

Drive C being processed.

The MIRROR process was successful.

Deletion-tracking software being installed.

The following drives are supported:
Drive C - Default files saved.

Installation complete.

Once you enable MIRROR, you won't need to enter the first letter of every filename as you undelete the files. For instance, if you'd issued the MIRROR command before deleting the files in the previous example and then issued the same UNDELETE command, DOS would have responded

```
Directory: C:\BATCH  
File Specifications: *.BAT
```

```
Deletion-tracking file contains  x deleted files.  
Of those,      x files have all clusters available,  
              0 files have some clusters available,  
              0 files have no clusters available.
```

```
MS-DOS directory contains  x deleted files.  
Of those,      x files may be recovered.
```

Using the deletion-tracking file.

```
filename BAT size date time ...attrib Deleted: date time  
All of the clusters for this file are available. Undelete  
(Y/N)?
```

At this prompt, you press Y, and DOS responds

```
File successfully undeleted.
```

Then, DOS displays information for the next file and asks if you want to undelete the file. DOS repeats this sequence for each recoverable file. Once you've responded to the prompt for each file, DOS returns you to the command prompt.

Sentry protection before deleting the *.BAT files in our previous example, you could move them from the SENTRY directory (where they moved when you deleted them) to the C:\BATCH directory by typing

```
C:\>undelete c:\batch\*.bat
```

Again, this command displays UNDELETE's copyright information. Then it displays

```
Directory: C:\BATCH
File Specifications: *.BAT
```

```
Delete Sentry control file contains    x deleted files.

Deletion-tracking file contains    0 deleted files.
Of those,    0 files have all clusters available,
            0 files have some clusters available,
            0 files have no clusters available.
```

```
MS-DOS directory contains    x deleted files.
Of those,    0 files may be recovered.
```

Using the Delete Sentry method.

```
filename BAT size date time ...attrib Deleted: date time
This file can be 100% undeleted. Undelete (Y/N)?
```

At this prompt, you press Y, and DOS responds

```
File successfully undeleted.
```

Then, DOS displays information for the next file and asks if you want to undelete the file. DOS repeats this sequence for each recoverable file. Once you've responded to the prompt for each file, DOS returns you to the command prompt.

Like Delete Tracker mode, Delete Sentry requires 13.5 Kb of memory for the memory-resident portion of the UNDELETE utility. In addition, the SENTRY directory and its files can occupy up to 7% of your hard disk space. Once this limit is reached, UNDELETE purges the oldest files in the SENTRY directory to make room for the most recently deleted files.

A note on precedence

You might have noticed that regardless of which level of protection you've invoked, the basic syntax of the UNDELETE command is the same:

```
C:\>undelete drive:\path\filename
```

When you issue this command, it first checks to see if you've invoked Delete Sentry mode. If you have, then UNDELETE uses the Delete Sentry method to restore deleted files. If you haven't, the command checks for Delete Tracking mode and uses the Delete Tracking method if it finds the method. If you haven't invoked Delete Sentry or Delete Tracking, UNDELETE restores files by using the standard method. To find out what switches you can use to override this precedence, read "Other Switches You Can Use with UNDELETE," below. ■

Other switches you can use with UNDELETE

VERSION
5.0 & 6.0

In "DOS 6's UNDELETE Offers Three Levels of Protection from File Deletion" and "Undeleting Files in DOS 5," we demonstrate UNDELETE's main options. Here's a quick look at some other options you can select with UNDELETE:

SWITCH	VERSION	FUNCTION
/ALL	5 & 6	Recovers files without prompting for confirmation
/DOS	5 & 6	Recovers files by using the standard deletion protection method, even if higher protection levels are enabled
/DS	6	Recovers files by using the Delete Sentry method
/DT	5 & 6	Recovers files by using the Delete Tracker method (even if Delete Sentry protection is enabled in Version 6)
/LIST	5 & 6	Lists deleted files that can be recovered
/LOAD	6	Loads UNDELETE into memory
/PURGEdrive	6	Purges all files in the SENTRY directory of the specified drive
/STATUS	6	Displays the level of deletion protection enabled for each drive
/Tdrive -entries	6	Enables Delete Tracker protection for the specified drive while setting a limit on the number of entries in the PCTRACKR.DEL file

Check your PATH before running MemMaker

By David Reid

While DOS 6's MemMaker program will normally do a good job of optimizing your computer's memory, it may destroy some files in your DOS directory if you've configured your PATH environment variable improperly. In some cases, you may not be able to boot from your hard disk after these files become corrupt. In this article, we'll explain how you can recover from this situation. In addition, we'll explain how to safely use MemMaker to avoid this problem entirely.

The PATH's the problem

In a properly configured PATH environment variable, each path contains a drive designation as well as a directory name. For example, here's a properly configured PATH environment variable:

```
PATH=C:\;C:\DOS;C:\WINDOWS;C:\UTILS
```

Unfortunately, in an attempt to squeeze more paths into the PATH environment variable's 127 available characters, some users omit the drive designations from one or more paths. For instance, they might define PATH as

```
PATH=C:\;\DOS;\WINDOWS;\UTILS
```

There's a rumor going around that this type of definition for PATH works because the C: drive designation in the first path will remain in effect for all subsequent paths in PATH. However, this simply isn't true. For any path with no drive designation, DOS uses the current drive. Therefore, if the current drive is A: and you try to run the FORMAT command, DOS looks in A:\DOS—not C:\DOS—for a COM, EXE, or BAT file named FORMAT.

Before DOS 6, the worst that could happen when you configured your PATH environment variable in this manner was a *Bad command or file name* message. However, DOS 6's MemMaker program crashes when it finds \DOS instead of C:\DOS in the PATH environment variable.

The side effects

If the path for your DOS directory doesn't include drive specifications, you'll know something has gone wrong as soon as you run MemMaker. You'll see MemMaker's first screen, which explains the program's purpose and prompts you to press [Enter] to continue. Immediately after you press [Enter], MemMaker displays the message

and then the error message

```
Sector not found
Retry (Y/N)
```

Although the error message prompts you to press Y or N, neither keystroke elicits a response from the program. At this point, there's only one keystroke you can use to escape from the program—[Ctrl][Alt][Del].

When your computer reboots, you'll probably see a series of error messages because MemMaker has corrupted the following files in your DOS directory:

```
HIMEM.SYS
EMM386.EXE
CHKSTATE.SYS
MEMMAKER.HLP
MEMMAKER.EXE
SIZER.EXE
```

Your computer may hang during execution of CONFIG.SYS, depending on the drivers you load from your CONFIG.SYS file. If this occurs, you'll need to boot DOS 6 from drive A: and then repair the files in your DOS directory. Fortunately, MemMaker's damage is limited to these six files.

At this point, you can recover by reinstalling DOS 6 or by selectively restoring these six files from the DOS 6 installation diskettes. However, before you can safely run MemMaker, you must fix your PATH environment variable.

Avoiding the problem

To prevent MemMaker from corrupting files in your DOS directory, you should check your PATH environment variable before running MemMaker. To do so, type the command

```
PATH
```

at the DOS command-line prompt. In response, DOS will display the current PATH environment variable.

Locate the path for your DOS directory and make sure it contains a drive specification. If it doesn't, edit AUTOEXEC.BAT and change the line defining PATH so your DOS directory contains a drive letter prefix. Then, save AUTOEXEC.BAT and reboot so the change will take effect. Now you can safely run MemMaker. ■

Please wait while MemMaker copies files to your startup drive

David Reid is editor-in-chief of The DOS Authority, a Cobb Group publication for advanced DOS users.

DOS 6 won't lose your data

I got a call the other day from our County Extension Agent—yes, I really *do* live on a ranch in a rural county in Montana. He was concerned because his monthly computer newsletter from the state extension office at Montana State University cautioned against upgrading to DOS 6 because of stories about errors that caused loss of data, especially when using DoubleSpace. After assuring him the dangers seemed grossly exaggerated, I thought it would be appropriate to devote a column in *Inside DOS* to the same question.

Letters-to-the-editor columns in some magazines and the DOS forum on CompuServe and other online systems are carrying stories about DOS 6 that tell of damaged disks and lost data, most (but not all) associated with DoubleSpace. Frequently, the problems are said not to occur immediately, but to crop up a few days or weeks after installation, with no warning or apparent cause. Reformatting the hard disk is usually the only remedy, although in some instances even reformatting hasn't revitalized the disk.

Errors can't be reproduced

If true, these problems would warrant a recall, probably the most expensive in the history of the software industry because Microsoft has been shipping more than a million copies a month since DOS 6 was released. But in my experience, DOS 6 doesn't seem to have these problems. I started using pre-release (beta test) versions of DOS 6 in November 1992,

and I haven't had a significant problem since late January (there were three additional pre-release versions after that spot of trouble). I used pre-release versions on three different systems: an IBM PS/2 Model 80 (16 MHz 386) and two Gateway 2000 systems (33 MHz 386 and 33 MHz 486).

I have a variety of devices attached to these machines, including two different scanners, a CD-ROM drive, two different sound cards, a standard laser printer connected to the parallel port, and a LaserMaster high-resolution laser printer that uses its own interface card. None of these devices or their device drivers posed any problems for DOS 6. I further tempted fate by installing Artisoft's LANtastic network hardware and software to link the two Gateway 2000 systems using a pre-release version of DOS 6, and neither DOS nor LANtastic so much as blinked. Within the limits of my experience, Version 6 has proven to be rock solid.

Why all the stories?

So what about all these stories? Many of the reports may well be true, but it doesn't follow that DOS 6 is therefore seriously flawed. There are opportunities for pilot error when installing DOS 6, especially when tinkering with DoubleSpace, MemMaker, and some of the other new utilities. Some systems may have just the right combination of unusual hardware that won't let DOS 6 install properly or, if it does install, run properly. Some less-

Nothing—not even DoubleSpace—is forever

VERSION
6.0

The DOS 6 manual cautions that there's no turning back once you've compressed a file by using DoubleSpace, but that isn't quite true. DoubleSpace doesn't do anything magical to your hard disk—it just creates some hidden files and manages them in a special way. If, for some reason, you want to uninstall DoubleSpace, it takes only a few DOS commands to undo what DoubleSpace has done; the amount of preparatory work, however, depends on how full your hard disk is and how many files stored on your compressed disk you want to keep. **Caution:** You probably shouldn't try to uninstall DoubleSpace unless you're pretty familiar with DOS and you have a good reason to do it.

First, here's a quick review of what DoubleSpace does to increase the capacity of your hard disk:

- DoubleSpace creates a hidden file named DBLSPACE.000 to hold all your compressed files and manages the space in this file as if it were a hard disk.
- It also creates a new hard disk volume—D: or H: if your hard disk is drive C:—containing files that shouldn't be compressed, including DBLSPACE.000.

- When you install DoubleSpace, it adds a DEVICE command to CONFIG.SYS and copies a hidden file named DBLSPACE.BIN to the root directory of your hard disk. Even if you delete the DEVICE command from CONFIG.SYS, DoubleSpace still runs on your system until you delete or rename DBLSPACE.BIN.

When you refer to drive C: in a command after you install DoubleSpace on a system with one hard disk, you're actually referring to the contents of DBLSPACE.000—the large hidden file that contains all your compressed files. When you refer to drive D: (or H:), you're actually referring to the uncompressed files remaining on your hard disk.

Saving your files to uncompressed space

To remove DoubleSpace from your system and restore normal operation to your hard disk, you must delete *all* the files DoubleSpace created. First, however, you must back up all the compressed files you want to save to an uncompressed volume—drive D: or H: or a floppy disk. If your hard disk is less than half full, you can probably

common systems might have a ROM BIOS, device driver, disk controller, or other system software that creates a problem.

Microsoft tried to cover as many bases as possible with the beta test of DOS 6. More than 7,000 people participated in the test of the pre-release versions of DOS. In order to continue participating (and to receive their free copy of the released version), beta testers had to note the result of using a variety of DOS 6 features and run some programs that produced special reports on a disk and then send the report and disk back to Microsoft. This assured that DOS was, indeed, being installed on a wide variety of machines and assured a minimal level of testing. In practice, many testers substituted the pre-release versions of DOS 6 for whatever version had been installed and used the system for routine work, especially during the latter two or three pre-release versions—not a prudent strategy, perhaps, but less time consuming and one that yielded a thorough shakedown of at least the commonly used parts of DOS.

But despite the legions of beta testers, the variety of possible hardware combinations almost guarantees some problems. Firm numbers are elusive, but estimates put the number of PC-compatible computers somewhere between 75 million and 100 million. Factor in the different disk drives and controllers and video boards—not to mention the other peripheral devices that can be attached—and the combinations multiply to a staggering number. This smorgasbord of hardware makes the PC

a more appealing and economical platform than the Apple-only Macintosh, but it also creates the possibility of lurking incompatibilities that might account for some of the reported problems.

We love a good story

Because computer users frequently use modems to communicate via nationwide networks such as CompuServe, MCI Mail, or the Internet, their stories can spread quickly. Combine this human predilection for repeating fantastic stories with some equally human resentment toward a giant corporation (Microsoft), and you have the ingredients for blowing a few inevitable problems way out of proportion. So, have there been some problems with DOS 6? Yes, a few. But the documented cases of reproducible bugs are rare, and none have yet been found that pose a general threat. If you've upgraded to DOS 6, don't worry that you're placing your system in jeopardy—if things are OK so far, chances are overwhelmingly in your favor that nothing will happen. If you haven't upgraded yet and are trying to decide, make your decision based on whether you want the new features, not on rumors of serious bugs. The problems *are* greatly exaggerated. ■

Contributing Editor Van Wolverton is the author of the best-selling books Running MS-DOS and Supercharging MS-DOS. Van, who has worked for IBM and Intel, lives in Alberton, Montana.

reduce the size of DBLSPACE.000 (increasing the uncompressed space available on drive D: or H:) and use XCOPY to save the files you need.

To reduce the size of the compressed drive, start DoubleSpace by typing *dblspace*; then, choose Change Size... from the Drive menu. You'll see a dialog box listing the current sizes of and free space on the compressed and uncompressed drives. You'll also see minimum and maximum limits on free space for each drive. The cursor will be in the New Free Space field in the Uncompressed Drive column. Enter a value in this field that's somewhat larger than the value in the Minimum Free Space field in the Compressed Drive column and then press [Enter]. DoubleSpace tells you it's changing the size of drive C: and then that it's remounting drive C:.

When control returns, exit DoubleSpace by choosing Exit from the Drive menu. Now use XCOPY to save all the files on drive C: you want to keep (save at least the DOS files and the root directory). If you have room for all the compressed files and the root directory is current, you can simply type *xcopy *.* D: /S* in order to make an uncompressed copy of every file on the disk. (Substitute H for D if your uncompressed drive is H:.) When you've finished, you're ready to delete DoubleSpace.

If you don't have enough space on your hard disk to hold uncompressed versions of the files you want to save, you'll have to back them up to floppy disks by using MSBackup or XCOPY. When you've backed up the files, you're ready to delete DoubleSpace.

Deleting DoubleSpace from your system

To get rid of DoubleSpace, you must delete the DEVICE command DoubleSpace puts in CONFIG.SYS and all the files it creates. First, type *edit config.sys* to open the CONFIG.SYS file in the DOS Editor. Then, delete the line *DEVICE=C:\DOS\DBLSPACE.SYS*. (It might use the DEVICEHIGH command instead, and it might include some parameters added by MemMaker, such as */L:2,52272*.) After deleting this line, exit the DOS Editor by pressing [Alt]F,X. When the prompt to save the file appears, press Y.

Next, type *attrib -s -h -r d:\dblspace.** and press [Enter] to make all the DoubleSpace files accessible. Then, type *del d:\dblspace.** and press [Enter] to delete these files.

Now press [Ctrl][Alt][Del] to restart your system. When DOS displays the command prompt, type *chkdsk* and check the bytes available; you should be back to uncompressed disk operations.

Microsoft Technical Support
(206) 454-2030

Please include account number from label with any correspondence.

Tell us about yourself

As publishers of *Inside DOS*, The Cobb Group wants to tailor the journal to meet your needs as closely as possible. Please help us get to know you better by taking a few moments to fill out this short survey. Just photocopy this page, fill it out, and mail or fax it to us by August 30. Thanks for your input!

Return surveys to or fax the completed form to
Inside DOS (502) 491-4200
The Cobb Group
P.O. Box 35160
Louisville, KY 40232

DOS USER PROFILE

- How would you rate your familiarity with DOS?
☐ Novice ☐ Experienced ☐ Expert
- How would you rate your proficiency with batch files?
☐ Novice ☐ Experienced ☐ Expert
- How would you rate your proficiency with QBasic?
☐ Novice ☐ Experienced ☐ Expert
- Do you use the DOS Shell?
☐ Never ☐ Sometimes ☐ Often
- Do you use Windows?
☐ Never ☐ Sometimes ☐ Often
- Do you use third-party utilities?
☐ Never ☐ Sometimes ☐ Often
- Have you updated to DOS 6?
☐ Yes ☐ No, but I plan to ☐ No, I don't plan to
- Do you take advantage of MemMaker in DOS 6?
☐ Yes ☐ No ☐ N/A
- Do you use DoubleSpace in DOS 6?
☐ Yes ☐ No ☐ N/A

JOURNAL TOPICS

10. Tell us what kinds of articles you'd like to see (with 1 indicating those you least want and 5 those you most want)

	LEAST					MOST				
Short tips and DOS shortcuts	1	2	3	4	5					
Advanced technical articles	1	2	3	4	5					
Tutorial articles on individual functions	1	2	3	4	5					
Batch file articles and tips	1	2	3	4	5					
Using QBasic	1	2	3	4	5					
Developer issues	1	2	3	4	5					
Reviews of DOS utilities	1	2	3	4	5					
Windows tips	1	2	3	4	5					
Network tips	1	2	3	4	5					
Other _____										

TELL US ABOUT YOUR COMPUTER

- What kind of CPU do you have?
☐ 286 ☐ 386 ☐ 486
- How much RAM do you have?
☐ 4 Mb ☐ 5 to 8 Mb ☐ More than 8 Mb
- What kind of video display do you have?
☐ EGA ☐ VGA ☐ SuperVGA Other _____
- How big is your hard drive?
☐ Less than 40 Mb ☐ 40 to 100 Mb
☐ 100 to 400 Mb ☐ Greater than 400 Mb
- What kind of printer(s) do you use?
☐ Dot-matrix ☐ Laser ☐ Both ☐ Other _____
- Do you use a mouse or other pointing device?
☐ Yes ☐ No
- Are you running DOS from a network?
☐ Yes ☐ No
- Do you read other Cobb Group publications?
☐ Yes ☐ No

Which ones? _____

